

Val

001
001
001
001
001
001
001
001
001
001
7FF

SSSSSSSSSSSS	MMM	MMM	GGGGGGGGGGGG	RRRRRRRRRRRR	TTTTTTTTTTTTTT	LLL
SSSSSSSSSSSS	MMM	MMM	GGGGGGGGGGGG	RRRRRRRRRRRR	TTTTTTTTTTTTTT	LLL
SSSSSSSSSSSS	MMM	MMM	GGGGGGGGGGGG	RRRRRRRRRRRR	TTTTTTTTTTTTTT	LLL
SSS	MMMMMM	MMMMMM	GGG	RRR	RRR	TTT
SSS	MMMMMM	MMMMMM	GGG	RRR	RRR	TTT
SSS	MMMMMM	MMMMMM	GGG	RRR	RRR	TTT
SSS	MM	MM	GGG	RRR	RRR	TTT
SSS	MM	MM	GGG	RRR	RRR	TTT
SSS	MM	MM	GGG	RRR	RRR	TTT
SSS	MM	MM	GGG	RRR	RRR	TTT
SSS	SSSSSS	MM	MM	GGG	RRRRRRRRRR	TTT
SSS	SSSSSS	MM	MM	GGG	RRRRRRRRRR	TTT
SSS	SSSSSS	MM	MM	GGG	RRRRRRRRRR	TTT
SSS	SSS	MM	MM	GGG	GGGGGGGG	RRR RRR
SSS	SSS	MM	MM	GGG	GGGGGGGG	RRR RRR
SSS	SSS	MM	MM	GGG	GGGGGGGG	RRR RRR
SSS	SSS	MM	MM	GGG	GGGGGGGG	RRR RRR
SSS	SSS	MM	MM	GGG	GGGGGGGG	RRR RRR
SSS	SSS	MM	MM	GGG	GGGGGGGG	RRR RRR
SSS	SSS	MM	MM	GGG	GGGGGGGG	RRR RRR
SSSSSSSSSS	MM	MM	GGGGGGGG	RRR	RRR	TTT
SSSSSSSSSS	MM	MM	GGGGGGGG	RRR	RRR	TTT
SSSSSSSSSS	MM	MM	GGGGGGGG	RRR	RRR	TTT

FILEID**SMGPRVNP

110

1

```
0001 0 XTITLE 'SMGSSPRVINP - Private Input support routines'
0002 0 MODULE SMGSSPRVINP (
0003 0 IDENT = '1-001' : File: SMGPRVINP.B32 Edit: STAN1001
0004 0 )
0005 1 BEGIN
0006 1
0007 1 ****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *
0029 1 ****
```

31 0030 1 ++
32 0031 1 FACILITY: Screen Management
33 0032 1
34 0033 1 ABSTRACT:
35 0034 1 The procedures in this module act as interfaces between the
36 0035 1 virtual displays and pasteboards and associated data structures, and
37 0036 1 the keyboard input side of SMG. These routines are called to pass
38 0037 1 information about changes to the physical screen that have been
39 0038 1 brought about by input-related activities.
40 0039 1 These are in a module by themselves so that they can be explicitly
41 0040 1 linked into programs that need them (like WHAT) since they
42 0041 1 will not be in the SMGSHR vector.
43 0042 1
44 0043 1 ENVIRONMENT: User mode, Shared library routines.
45 0044 1
46 0045 1 AUTHOR: R. Reichert, CREATION DATE: 9-Mar-1983
47 0046 1
48 0047 1
49 0048 1
50 0049 1
51 0050 1 MODIFIED BY:
1-001 - Original. Moved out of other modules. STAN 7-Mar-1984.
!--

```
53 0051 1 %SBTTL 'Declarations'  
54 0052 1  
55 0053 1 SWITCHES:  
56 0054 1  
57 0055 1  
58 0056 1  
59 0057 1 LINKAGES:  
60 0058 1  
61 0059 1 NONE  
62 0060 1  
63 0061 1 TABLE OF CONTENTS:  
64 0062 1  
65 0063 1  
66 0064 1 FORWARD ROUTINE  
67 0065 1  
68 0066 1 ! Private entry points:  
69 0067 1  
70 0068 1 SMG$INVALIDATE_DISPLAY, ! Mark contents of display as unknown  
71 0069 1  
72 0070 1 SMG$REPORT_CHANGE_INSERT, ! Report change to physical  
73 0071 1 screen in insert mode.  
74 0072 1  
75 0073 1 SMG$REPORT_CHANGE_REPLACE; ! Report change to physical  
76 0074 1 screen in replace mode.  
77 0075 1  
78 0076 1  
79 0077 1 INCLUDE FILES  
80 0078 1  
81 0079 1  
82 0080 1 REQUIRE 'RTLIN:SMGPROLOG'; ! defines psects, macros, tcb,  
83 0158 1 wcb, & terminal symbols  
84 0159 1  
85 0160 1  
86 0161 1 EXTERNAL REFERENCES  
87 0162 1  
88 0163 1  
89 0164 1 EXTERNAL  
90 0165 1 PBD_L_COUNT, ! No. of pasteboards we currently have  
91 0166 1  
92 0167 1 PBD_A_PBCB : VECTOR [PBD_K_MAX_PB, LONG],  
93 0168 1 ! Table of addresses of PBCB's  
94 0169 1  
95 0170 1 PBD_V_PB_AVAIL : BITVECTOR [PBD_K_MAX_PB];  
96 0171 1 ! Bit vector of pasteboard id numbers in use.  
97 0172 1  
98 0173 1 EXTERNAL ROUTINE  
99 0174 1 LIB$GET_VM. ! Allocate heap storage  
100 0175 1  
101 0176 1 SMG$INSERT_CHARS, ! Insert char in virtual display buffer  
102 0177 1 ! and cause output.  
103 0178 1  
104 0179 1 SMG$FILL_WINDOW_BUFFER, ! Map all virtual display buffers to  
105 0180 1 ! the window buffer for a given PBCB  
106 0181 1  
107 0182 1 SMG$FLUSH_BUFFER, ! Flush any pending output to terminal  
108 0183 1  
109 0184 1 SMG$FORCE_SCROLL_REG, ! Force scroll region to specified
```

```
: 110      0185 1           ! lines.
: 111      0186 1
: 112      0187 1           SMGSSLOCATE_PP.    ! Locate pasting packet that joins a virtual
: 113      0188 1           display to a pasteboard.
: 114      0189 1
: 115      0190 1           SMGSSMOVE_TEXT_TO_SCREEN_BUF.
: 116      0191 1
: 117      0192 1           SMGSSMOVE_TEXT_TO_WINDOW_BUF.    ! Map single virtual display to
: 118      0193 1           window buffer.
: 119      0194 1
: 120      0195 1           SMGSSOCCLUE.      ! Determine overlap between two rectangular
: 121      0196 1           areas.
: 122      0197 1           SMGSSMIN_UPD.      ! Minimum output routine
: 123      0198 1
: 124      0199 1           SMGSSPUT_TEXT_TO_BUFFER;    ! Text to virtual display buffer
: 125      0200 1
: 126      0201 1           EXTERNAL LITERAL
: 127      0202 1
: 128      0203 1           SMGS_FATERRLIB.    ! Fatal error in library procedure
: 129      0204 1           SMGS_INVARARG.    ! Invalid argument
: 130      0205 1           SMGS_INVCOL.      ! Invalid column number
: 131      0206 1           SMGS_INVDIS_ID.    ! Invalid virtual display id
: 132      0207 1           SMGS_INVPAS_ID.    ! Invalid pasteboard id
: 133      0208 1           SMGS_INVROW;      ! Invalid row number
```

135 0209 1 %SBTTL 'SMG\$INVALIDATE_DISPLAY - Mark display as being privately used'
136 0210 1 GLOBAL ROUTINE SMG\$INVALIDATE_DISPLAY (DISPLAY_ID) =
137 0211 1
138 0212 1 ++
139 0213 1 FUNCTIONAL DESCRIPTION:
140 0214 1
141 0215 1 This routine is called when ever a change has been completed
142 0216 1 to a given virtual display and the user had previously
143 0217 1 written into that display on his own, without using
144 0218 1 SMG routines.
145 0219 1
146 0220 1 The virtual display must not be occluded.
147 0221 1
148 0222 1 Each pasteboard to which this display is pasted is isolated and
149 0223 1 its window image must be redrawn.
150 0224 1 The new physical pasteboard cursor position gets set.
151 0225 1
152 0226 1 CALLING SEQUENCE:
153 0227 1
154 0228 1 ret_status.wlc.v = SMG\$INVALIDATE_DISPLAY (DISPLAY_ID.rl.)
155 0229 1
156 0230 1 FORMAL PARAMETERS:
157 0231 1
158 0232 1 DISPLAY_ID.rl.r Display ID of virtual display.
159 0233 1
160 0234 1
161 0235 1
162 0236 1
163 0237 1
164 0238 1
165 0239 1
166 0240 1
167 0241 1
168 0242 1
169 0243 1
170 0244 1
171 0245 1
172 0246 1
173 0247 1
174 0248 1
175 0249 1 IMPLICIT INPUTS:
176 0250 1
177 0251 1
178 0252 1
179 0253 1
180 0254 1
181 0255 1
182 0256 1
183 0257 1
184 0258 1
185 0259 1
186 0260 1
187 0261 1
188 0262 1
189 0263 1
190 0264 1
191 0265 1
192 0266 1
193 0267 1
194 0268 1
195 0269 1
196 0270 1
197 0271 1
198 0272 1
199 0273 1
200 0274 1
201 0275 1
202 0276 1
203 0277 1
204 0278 1
205 0279 1
206 0280 1
207 0281 1
208 0282 1
209 0283 1
210 0284 1
211 0285 1
212 0286 1
213 0287 1
214 0288 1
215 0289 1
216 0290 1
217 0291 1
218 0292 1
219 0293 1
220 0294 1
221 0295 1
222 0296 1
223 0297 1
224 0298 1
225 0299 1
226 0300 1
227 0301 1
228 0302 1
229 0303 1
230 0304 1
231 0305 1
232 0306 1
233 0307 1
234 0308 1
235 0309 1
236 0310 1
237 0311 1
238 0312 1
239 0313 1
240 0314 1
241 0315 1
242 0316 1
243 0317 1
244 0318 1
245 0319 1
246 0320 1
247 0321 1
248 0322 1
249 0323 1
250 0324 1
251 0325 1
252 0326 1
253 0327 1
254 0328 1
255 0329 1
256 0330 1
257 0331 1
258 0332 1
259 0333 1
260 0334 1
261 0335 1
262 0336 1
263 0337 1
264 0338 1
265 0339 1
266 0340 1
267 0341 1
268 0342 1
269 0343 1
270 0344 1
271 0345 1
272 0346 1
273 0347 1
274 0348 1
275 0349 1
276 0350 1
277 0351 1
278 0352 1
279 0353 1
280 0354 1
281 0355 1
282 0356 1
283 0357 1
284 0358 1
285 0359 1
286 0360 1
287 0361 1
288 0362 1
289 0363 1
290 0364 1
291 0365 1
292 0366 1
293 0367 1
294 0368 1
295 0369 1
296 0370 1
297 0371 1
298 0372 1
299 0373 1
300 0374 1
301 0375 1
302 0376 1
303 0377 1
304 0378 1
305 0379 1
306 0380 1
307 0381 1
308 0382 1
309 0383 1
310 0384 1
311 0385 1
312 0386 1
313 0387 1
314 0388 1
315 0389 1
316 0390 1
317 0391 1
318 0392 1
319 0393 1
320 0394 1
321 0395 1
322 0396 1
323 0397 1
324 0398 1
325 0399 1
326 0400 1
327 0401 1
328 0402 1
329 0403 1
330 0404 1
331 0405 1
332 0406 1
333 0407 1
334 0408 1
335 0409 1
336 0410 1
337 0411 1
338 0412 1
339 0413 1
340 0414 1
341 0415 1
342 0416 1
343 0417 1
344 0418 1
345 0419 1
346 0420 1
347 0421 1
348 0422 1
349 0423 1
350 0424 1
351 0425 1
352 0426 1
353 0427 1
354 0428 1
355 0429 1
356 0430 1
357 0431 1
358 0432 1
359 0433 1
360 0434 1
361 0435 1
362 0436 1
363 0437 1
364 0438 1
365 0439 1
366 0440 1
367 0441 1
368 0442 1
369 0443 1
370 0444 1
371 0445 1
372 0446 1
373 0447 1
374 0448 1
375 0449 1
376 0450 1
377 0451 1
378 0452 1
379 0453 1
380 0454 1
381 0455 1
382 0456 1
383 0457 1
384 0458 1
385 0459 1
386 0460 1
387 0461 1
388 0462 1
389 0463 1
390 0464 1
391 0465 1
392 0466 1
393 0467 1
394 0468 1
395 0469 1
396 0470 1
397 0471 1
398 0472 1
399 0473 1
400 0474 1
401 0475 1
402 0476 1
403 0477 1
404 0478 1
405 0479 1
406 0480 1
407 0481 1
408 0482 1
409 0483 1
410 0484 1
411 0485 1
412 0486 1
413 0487 1
414 0488 1
415 0489 1
416 0490 1
417 0491 1
418 0492 1
419 0493 1
420 0494 1
421 0495 1
422 0496 1
423 0497 1
424 0498 1
425 0499 1
426 0500 1
427 0501 1
428 0502 1
429 0503 1
430 0504 1
431 0505 1
432 0506 1
433 0507 1
434 0508 1
435 0509 1
436 0510 1
437 0511 1
438 0512 1
439 0513 1
440 0514 1
441 0515 1
442 0516 1
443 0517 1
444 0518 1
445 0519 1
446 0520 1
447 0521 1
448 0522 1
449 0523 1
450 0524 1
451 0525 1
452 0526 1
453 0527 1
454 0528 1
455 0529 1
456 0530 1
457 0531 1
458 0532 1
459 0533 1
460 0534 1
461 0535 1
462 0536 1
463 0537 1
464 0538 1
465 0539 1
466 0540 1
467 0541 1
468 0542 1
469 0543 1
470 0544 1
471 0545 1
472 0546 1
473 0547 1
474 0548 1
475 0549 1
476 0550 1
477 0551 1
478 0552 1
479 0553 1
480 0554 1
481 0555 1
482 0556 1
483 0557 1
484 0558 1
485 0559 1
486 0560 1
487 0561 1
488 0562 1
489 0563 1
490 0564 1
491 0565 1
492 0566 1
493 0567 1
494 0568 1
495 0569 1
496 0570 1
497 0571 1
498 0572 1
499 0573 1
500 0574 1
501 0575 1
502 0576 1
503 0577 1
504 0578 1
505 0579 1
506 0580 1
507 0581 1
508 0582 1
509 0583 1
510 0584 1
511 0585 1
512 0586 1
513 0587 1
514 0588 1
515 0589 1
516 0590 1
517 0591 1
518 0592 1
519 0593 1
520 0594 1
521 0595 1
522 0596 1
523 0597 1
524 0598 1
525 0599 1
526 0500 1
527 0501 1
528 0502 1
529 0503 1
530 0504 1
531 0505 1
532 0506 1
533 0507 1
534 0508 1
535 0509 1
536 0500 1
537 0501 1
538 0502 1
539 0503 1
540 0504 1
541 0505 1
542 0506 1
543 0507 1
544 0508 1
545 0509 1
546 0500 1
547 0501 1
548 0502 1
549 0503 1
550 0504 1
551 0505 1
552 0506 1
553 0507 1
554 0508 1
555 0509 1
556 0500 1
557 0501 1
558 0502 1
559 0503 1
560 0504 1
561 0505 1
562 0506 1
563 0507 1
564 0508 1
565 0509 1
566 0500 1
567 0501 1
568 0502 1
569 0503 1
570 0504 1
571 0505 1
572 0506 1
573 0507 1
574 0508 1
575 0509 1
576 0500 1
577 0501 1
578 0502 1
579 0503 1
580 0504 1
581 0505 1
582 0506 1
583 0507 1
584 0508 1
585 0509 1
586 0500 1
587 0501 1
588 0502 1
589 0503 1
590 0504 1
591 0505 1
592 0506 1
593 0507 1
594 0508 1
595 0509 1
596 0500 1
597 0501 1
598 0502 1
599 0503 1
600 0504 1
601 0505 1
602 0506 1
603 0507 1
604 0508 1
605 0509 1
606 0500 1
607 0501 1
608 0502 1
609 0503 1
610 0504 1
611 0505 1
612 0506 1
613 0507 1
614 0508 1
615 0509 1
616 0500 1
617 0501 1
618 0502 1
619 0503 1
620 0504 1
621 0505 1
622 0506 1
623 0507 1
624 0508 1
625 0509 1
626 0500 1
627 0501 1
628 0502 1
629 0503 1
630 0504 1
631 0505 1
632 0506 1
633 0507 1
634 0508 1
635 0509 1
636 0500 1
637 0501 1
638 0502 1
639 0503 1
640 0504 1
641 0505 1
642 0506 1
643 0507 1
644 0508 1
645 0509 1
646 0500 1
647 0501 1
648 0502 1
649 0503 1
650 0504 1
651 0505 1
652 0506 1
653 0507 1
654 0508 1
655 0509 1
656 0500 1
657 0501 1
658 0502 1
659 0503 1
660 0504 1
661 0505 1
662 0506 1
663 0507 1
664 0508 1
665 0509 1
666 0500 1
667 0501 1
668 0502 1
669 0503 1
670 0504 1
671 0505 1
672 0506 1
673 0507 1
674 0508 1
675 0509 1
676 0500 1
677 0501 1
678 0502 1
679 0503 1
680 0504 1
681 0505 1
682 0506 1
683 0507 1
684 0508 1
685 0509 1
686 0500 1
687 0501 1
688 0502 1
689 0503 1
690 0504 1
691 0505 1
692 0506 1
693 0507 1
694 0508 1
695 0509 1
696 0500 1
697 0501 1
698 0502 1
699 0503 1
700 0504 1
701 0505 1
702 0506 1
703 0507 1
704 0508 1
705 0509 1
706 0500 1
707 0501 1
708 0502 1
709 0503 1
710 0504 1
711 0505 1
712 0506 1
713 0507 1
714 0508 1
715 0509 1
716 0500 1
717 0501 1
718 0502 1
719 0503 1
720 0504 1
721 0505 1
722 0506 1
723 0507 1
724 0508 1
725 0509 1
726 0500 1
727 0501 1
728 0502 1
729 0503 1
730 0504 1
731 0505 1
732 0506 1
733 0507 1
734 0508 1
735 0509 1
736 0500 1
737 0501 1
738 0502 1
739 0503 1
740 0504 1
741 0505 1
742 0506 1
743 0507 1
744 0508 1
745 0509 1
746 0500 1
74

```
177 0250 2 BEGIN
178 0251 2
179 0252 2 LOCAL
180 0253 2
181 0254 2 DCB : REF BLOCK [,BYTE],      ! Addr of display control block
182 0255 2 CURR_PP : REF BLOCK [,BYTE],    ! Addr of pasting packet under
183 0256 2                                inspection
184 0257 2
185 0258 2 STATUS; ! Status of subroutine calls
186 0259 2
187 0260 2 !+
188 0261 2 | This routine is independent of buffering.
189 0262 2 !-
190 0263 2
191 0264 2 $SMG$GET_DCBlock (.DISPLAY_ID, DCB);      ! Get DCB address
192 0265 2
193 0266 2 CURR_PP = .DCB [DCB_A_PP_NEXT];      ! Start of chain of pasting
194 0267 2                                packets to which this display
195 0268 2                                is pasted.
196 0269 2
197 0270 2 !+
198 0271 2 | Deal with each pasteboard that this virtual display is pasted to...
199 0272 2 !-
200 0273 2
201 0274 2 WHILE .CURR_PP NEQ DCB [DCB_A_PP_NEXT]
202 0275 2 DO
203 0276 3 BEGIN ! Overall loop
204 0277 3
205 0278 3 LOCAL
206 0279 3 PBCB : REF BLOCK [,BYTE],      ! Address of pasteboard control
207 0280 3 WCB : REF BLOCK [,BYTE],      ! Address of window control block
208 0281 3 TO_INDEX;                  ! Index into destination
209 0282 3
210 0283 3 PBCB = .CURR_PP [PP_A_PBCB_ADDR]; ! Select this pasteboard and WCB
211 0284 3 WCB = .PBCB [PBCB_A_WCB];      ! whose window needs rebuilding.
212 0285 3
213 0286 3 TO_INDEX = .CURR_PP [PP_W_TO_INDEX];
214 0287 3
215 0288 3 INCR R FROM 1 TO .CURR_PP [PP_W_ROWS_TO_MOVE]
216 0289 3 DO
217 0290 4 BEGIN ! For all rows in this display
218 0291 4 !+
219 0292 4 | Zero out the display buffer.
220 0293 4 !-
221 0294 4 CHSFILL (0,
222 0295 4     .CURR_PP [PP_W_MOVE_LENGTH],
223 0296 4     .WCB [WCB_A_SCR_TEXT_BUF] + .TO_INDEX);
224 0297 4
225 0298 4 TO_INDEX = .TO_INDEX + .WCB [WCB_W_NO_COLS];
226 0299 3 END; ! For all rows to move
227 0300 3
228 0301 3 CURR_PP = .CURR_PP [PP_A_NEXT_DCBlock]; ! Walk the DCB side of
229 0302 3                                the chain from front
230 0303 3                                to back.
231 0304 2 END; ! Overall loop
232 0305 2
233 0306 2 RETURN SSS_NORMAL
```

: 234
: 2350307 2
0308 1 END;

! End of routine SMG\$INVALIDATE_DISPLAY

				OFFC 00000
04	50 BC	04 38	BC A0	D0 00002
			06	12 00008
11	44	A0	91	0000D
			08	13 00011
	50 00000000G		8F	D0 00013 1\$:
			04	0001A
5A	04 BC	0001B	2\$:	MOVL
57	20 AA	0001F		ADISPLAY_ID, R0
50	20 AA	9E 00023	3\$:	CMPL 56(R0), ADISPLAY_ID
50	57 D1	00027		BNEQ 1\$
	2D 13	0002A		CMPB 68(R0), #17
50	14 A7	0002C		BEQL 2\$
56	08 A0	00030		MOVL #SMGS_INVDIS_ID, R0
58	20 A7	3C 00034		RET
58	1C A7	3C 00038		MOVL ADISPLAY_ID, DCB
		59 D4	0003C	MOVL 32(DCB), CURR_PP
		10 11	0003E	MOVAB 32(DCB), R0
22 A7	00	6E	00 2C	CMPL CURR_PP, R0
			00 00040	BEQL 6\$
		14 B648	00046	MOVL 20(CURR_PP), PBCB
		06	A6 3C	MOVL 8(PBCB), WCB
EC			00 00049	MOVZWL 32(CURR_PP), TO_INDEX
			50 C0	MOVZWL 28(CURR_PP), R1T
			0004D	CLRL R
		59 5B	F3 00050	BRB \$
		57 67	D0 00054	MOVC5 #0, (SP), #0, 34(CURR_PP), @20(WCB)-
			CA 11	[TO_INDEX]
			00057	MOVZWL 6(WCB), R0
		50 01	D0 00059	ADDL2 R0, TO_INDEX
			00059	AOBLEQ R1T, R-4\$
		04	0005C	MOVL (CURR_PP), CURR_PP

: Routine Size: 93 bytes, Routine Base: _SMG\$CODE + 0000

.TITLE SMG\$SPRVINP SMG\$SPRVINP - Private Input support routines
.IDENT \1-001\

.EXTRN PBD_L_COUNT, PBD_A_PBCB
.EXTRN PBD_V_PBAVAIL, [IB\$GET_VM
.EXTRN SMG\$INSERT_CHARS
.EXTRN SMG\$SFILL_WINDOW_BUFFER
.EXTRN SMG\$FLUSH_BUFFER
.EXTRN SMG\$FORCE_SCROLL_REG
.EXTRN SMG\$LOCATE_PP, SMG\$MOVE_TEXT_TO_SCREEN_BUF
.EXTRN SMG\$MOVE_TEXT_TO_WINDOW_BUF
.EXTRN SMG\$OCCLUDE, SMG\$MIN_UPD
.EXTRN SMG\$PUT_TEXT_TO_BUFFER
.EXTRN SMG\$FATERRLIB, SMG\$INVARG
.EXTRN SMG\$INVCOL, SMG\$INVDIS_ID
.EXTRN SMG\$INVPAS_ID, SMG\$INVROW

.PSECT _SMG\$CODE, NOWRT, SHR, PIC,2

.ENTRY SMG\$INVALIDATE_DISPLAY, Save R2,R3,R4,R5,- ; 0210
R6,R7,R8,R9,R10,R11

MOVL	ADISPLAY_ID, R0	0264
CMPL	56(R0), ADISPLAY_ID	
BNEQ	1\$	
CMPB	68(R0), #17	
BEQL	2\$	
MOVL	#SMGS_INVDIS_ID, R0	
RET		
MOVL	ADISPLAY_ID, DCB	0266
MOVL	32(DCB), CURR_PP	0274
MOVAB	32(DCB), R0	
CMPL	CURR_PP, R0	
BEQL	6\$	
MOVL	20(CURR_PP), PBCB	0283
MOVL	8(PBCB), WCB	0284
MOVZWL	32(CURR_PP), TO_INDEX	0286
MOVZWL	28(CURR_PP), R1T	0288
CLRL	R	0298
BRB	\$	
MOVC5	#0, (SP), #0, 34(CURR_PP), @20(WCB)-	0296
	[TO_INDEX]	
MOVZWL	6(WCB), R0	0298
ADDL2	R0, TO_INDEX	
AOBLEQ	R1T, R-4\$	0288
MOVL	(CURR_PP), CURR_PP	0301
BRB	3\$	0274
MOVL	#1, R0	0306
RET		0308

```
237 0309 1 %SBTTL 'SMG$$_REPORT_CHANGE_INSERT - Report change to screen -- insert'  
238 0310 1 GLOBAL ROUTINE SMG$$_REPORT_CHANGE_INSERT (  
239 0311 1 DISPLAY_ID,  
240 0312 1 PASTEBOARD_ID,  
241 0313 1 CHANGED_CHAR,  
242 0314 1 CHANGED_ROW,  
243 0315 1 CHANGED_COL,  
244 0316 1 TERMINATING_CHAR
```

FUNCTIONAL DESCRIPTION:

This routine is called to report a change made to the physical screen via an input action involving insertion. The character identified by CHANGED_CHAR has been positioned on the screen in a position corresponding to the virtual display coordinates CHANGED_ROW and CHANGED_COL.

`TERMINATING_CHAR` specifies the details of the change.

If TERMINATING_CHAR has not been specified, then CHANGED_CHAR has been echoed in the position indicated. The original contents of the line from CHANGED_COL to the right-hand edge of the virtual display need to be shifted to the right by one character. The new contents of CHANGED_COL+1 through the right-hand edge need to be redisplayed on the screen. Current cursor position is changed to reflect that one beyond that indicated by CHANGED_COL. If this new position is now beyond the bounds of the virtual display's dimensions then ?????.

If TERMINATING_CHAR was specified, CHANGED_CHAR has been echoed in the position indicated. Furthermore, this character was followed by a TERMINATING_CHAR. The original contents of the line from CHANGED_COL to the right-hand edge of the virtual display need to be shifted to the right by one character. The new contents of CHANGED_COL+1 through the right-hand edge need to be redisplayed on the screen.

Current cursor position is changed to reflect the effects of the TERMINATING_CHAR. If this new position is now beyond the bounds of the virtual display's dimensions then ????.

CALLING SEQUENCE:

```
ret_status.wlc.v = SMG$$_REPORT_CHANGE_INSERT (  
    DISPLAY_ID.rl.r,  
    PASTEBOARD_ID.rl.r,  
    CHANGED_CHAR.rb.r,  
    CHANGED_ROW.rl.r,  
    CHANGED_COL.rl.r  
    [.TERMINATING_CHAR.rb.r])
```

FORMAL PARAMETERS:

DISPLAY_ID.rl.r Display id of virtual display.
PASTEBOARD_ID.rl.r Pasteboard id.

```

294 0366 1 | CHANGED_CHAR.rb.r The character that modified the screen.
295 0367 1 |
296 0368 1 | CHANGED_ROW.rl.r Row number within the virtual display in
297 0369 1 | which CHANGED_CHAR was written.
298 0370 1 |
299 0371 1 | CHANGED_COL.rl.r Column number within the virtual display
300 0372 1 | where CHANGED_CHAR was written.
301 0373 1 |
302 0374 1 | [,TERMINATING_CHAR.rb.r] [Optional].
303 0375 1 | If supplied, the terminating character
304 0376 1 | that followed CHANGED_CHAR (See
305 0377 1 | functional description for meaning).
306 0378 1 |
307 0379 1 | IMPLICIT INPUTS:
308 0380 1 | NONE
309 0381 1 |
310 0382 1 | IMPLICIT OUTPUTS:
311 0383 1 | NONE
312 0384 1 |
313 0385 1 | COMPLETION STATUS:
314 0386 1 |
315 0387 1 | 0389 1 | SSS_NORMAL Normal successful completion
316 0388 1 | 0390 1 | SMGS_INVDIS_ID Invalid Display Id
317 0389 1 | 0391 1 | SMGS_INVPAS_ID Invalid Pasteboard Id
318 0390 1 | 0392 1 | SMGS_INVROW Invalid row specified
319 0391 1 | 0393 1 | SMGS_INVCOL Invalid column specified
320 0392 1 |
321 0393 1 |
322 0394 1 |
323 0395 1 | SIDE EFFECTS:
324 0396 1 |
325 0397 1 | NONE
326 0398 1 |
327 0399 1 |
328 0400 2 | BEGIN
329 0401 2 | BUILTIN
330 0402 2 | NULLPARAMETER;
331 0403 2 |
332 0404 2 | LOCAL
333 0405 2 | DESC : BLOCK [8,BYTE]. | Local descriptor
334 0406 2 | STATUS, | Status of subroutine calls
335 0407 2 | DCB : REF $DCB DECL, | Addr of display control block
336 0408 2 | PBCB : REF $PBCB DECL, | Addr of pasteboard control
337 0409 2 | PP : REF $PP DECL; | block.
338 0410 2 | | Addr of pasting packet
339 0411 2 |
340 0412 2 | SSMG$VALIDATE_ARGCOUNT (5, 6); | Test for right no. of args
341 0413 2 |
342 0414 2 | +
343 0415 2 | Get addresses of control blocks we need
344 0416 2 | -
345 0417 2 | SSMG$GET_DC B (.DISPLAY_ID, DCB); | Get DCB addr.
346 0418 2 | SSMG$GET_PBCB (.PASTEBOARD_ID, PBCB); | Get PBCB addr.
347 0419 3 | IF NOT (STATUS = SMG$LOCATE_PP (.DCB, .PBCB, PP)) | Get PP addr.
348 0420 2 | THEN
349 0421 2 | RETURN (.STATUS);
350 0422 2 |

```

```
351 0423 2     $SMG$VALIDATE_ROW_COL(..CHANGED_ROW, ..CHANGED_COL); ! Valid Pos.?
352 0424 2
353 0425 2
354 0426 2     !+ Initialize our local descriptor to point to the changed character.
355 0427 2     !-
356 0428 2     DESC [DSC$W_LENGTH] = 1;
357 0429 2     DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
358 0430 2     DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
359 0431 2     DESC [DSC$A_POINTER] = .CHANGED_CHAR;
360 0432 2
361 0433 2
362 0434 2     !+ Reflect this change in the virtual display text and attribute buffer,
363 0435 2     including new virtual display cursor position.
364 0436 2     Data from the affected column to the last-1 column of this line need
365 0437 2     to be moved one character position to the right and the changed
366 0438 2     character inserted at the indicated position. The attributes for the
367 0439 2     moved bytes must be moved as well.
368 0440 2     !-
369 0441 3     IF NOT (STATUS = SMG$INSERT_CHARS ( .DISPLAY_ID,
370 0442 3             .CHANGED_ROW,
371 0443 3             .CHANGED_COL,
372 0444 3             DESC))
373 0445 2     THEN
374 0446 2         RETURN (.STATUS);
375 0447 2
376 0448 2     IF NOT NULLPARAMETER (6)
377 0449 2     THEN
378 0450 3         BEGIN ! Terminator supplied
379 0451 3         !+
380 0452 3         Inspect supplied terminator to determine effect on cursor
381 0453 3         position in virtual display.
382 0454 3
383 0455 3
384 0456 3     !+ **** For now pretend the terminator is a <CR>. ****
385 0457 3     !+ This needs to act like a <CR><LF> pair.
386 0458 3
387 0459 4     IF NOT (STATUS = SMG$PUT_TEXT_TO_BUFFER (
388 0460 4             .DCB,
389 0461 4             .DCB [DCB_B_DEF_VIDEO_ATTR],
390 0462 4             ),
391 0463 4             XREF (CR^8 + LF), ! <CR><LF>
392 0464 4             .DCB [DCB_B_DEF_CHAR_SET]))
393 0465 3     THEN
394 0466 3         RETURN (.STATUS);
395 0467 2     END; ! Terminator supplied
396 0468 2
397 0469 2
398 0470 2     !+ Reflect this change in the appropriate positions of the window screen
399 0471 2     text and attribute buffers, including new screen cursor position.
400 0472 2
401 0473 3     IF NOT (STATUS = SMG$MOVE_TEXT_TO_WINDOW_BUF ( .PP))
402 0474 2     THEN
403 0475 2         RETURN (.STATUS);
404 0476 2
405 0477 2
406 0478 2     !+ Record what has happened to screen buffer as well.
407 0479 2     !-
```

```
408 0480 3 IF NOT (STATUS = SMG$SMOVE_TEXT_TO_SCREEN_BUF (.PP))
409 0481 2 THEN
410 0482 2 RETURN (.STATUS);
411 0483 2
412 0484 2
413 0485 2 Must now force the changes to be output.
414 0486 2
415 0487 2 IF .PP [PP_W_ROWS_TO_MOVE] NEQ 0
416 0488 2 THEN
417 0489 3 BEGIN
418 0490 3 ! Assume damage confined to single row.
419 0491 3 PBCB [PBCB_W_FIRST_CHANGED_ROW] = ..[CHANGED_ROW];
420 0492 3 PBCB [PBCB_W_LAST_CHANGED_ROW] = ..[CHANGED_ROW];
421 0493 3
422 0494 3 Assume damage in row from given pos. to end of line
423 0495 3 PBCB [PBCB_W_FIRST_CHANGED_COL] = ..[CHANGED_COL];
424 0496 3 PBCB [PBCB_W_LAST_CHANGED_COL] = .PBCB [PBCB_W_WIDTH];
425 0497 2 END:
426 0498 2
427 0499 2 STATUS = SMG$SMIN_UPD (.PBCB);
428 0500 2
429 0501 2
430 0502 2 If this virtual display is pasted to pasteboards other than the one
431 0503 2 identified in the call list, these additional pasteboard's window
432 0504 2 buffers must be updated as well. For these additional pasteboards,
433 0505 2 the changed byte in addition to the shifted remainder of the line
434 0506 2 must be output -- since they did not receive the originally echoed
435 0507 2 character.
436 0508 2
437 0509 2 PP = .DCB [DCB_A_PP_NEXT]; ! 1st in chain
438 0510 2 WHILE .PP NEQ RCB [DCB_A_PP_NEXT] ! While any packets remain...
439 0511 2 DO
440 0512 3 BEGIN ! Loop through all pasting packets for this DCB
441 0513 3 LOCAL
442 0514 3 NEW_PBCB : REF $PBCB_DECL; ! PBCB being considered
443 0515 3
444 0516 3 NEW_PBCB = .PP [PP_A_PBCB_ADDR]; ! PBCB for this packet
445 0517 3 IF .NEW_PBCB NEQ .PBCB ! If this isn't the one we started with
446 0518 3 THEN
447 0519 4 BEGIN ! Needs to be output
448 0520 5 IF NOT (STATUS = SMG$SFILL_WINDOW_BUFFER (.NEW_PBCB))
449 0521 4 THEN
450 0522 4 RETURN (.STATUS);
451 0523 4
452 0524 4 IF .PP [PP_W_ROWS_TO_MOVE] NEQ 0
453 0525 4 THEN
454 0526 5 BEGIN
455 0527 5 ! Assume damage confined to single row.
456 0528 5 PBCB [PBCB_W_FIRST_CHANGED_ROW] = ..[CHANGED_ROW];
457 0529 5 PBCB [PBCB_W_LAST_CHANGED_ROW] = ..[CHANGED_ROW];
458 0530 5
459 0531 5 ! Assume damage in row from given pos. to end of line
460 0532 5 PBCB [PBCB_W_FIRST_CHANGED_COL] = ..[CHANGED_COL];
461 0533 5 PBCB [PBCB_W_LAST_CHANGED_COL] = .PBCB [PBCB_W_WIDTH];
462 0534 4 END:
463 0535 4
464 0536 5 IF NOT (STATUS = SMG$SMIN_UPD (.NEW_PBCB))
```

SMGSSPRVINP - Private Input support routines
SMGSSREPORT_CHANGE_INSERT - Report change to sc

L 11
16-Sep-1984 01:10:09
14-Sep-1984 13:09:59
VAX-11 Bliss-32 V4.0-742
[SMGRTL.SRC]SMGPRVINP.B32;1

465 0537 4
466 0538 4
467 0539 3
468 0540 3
469 0541 3
470 0542 2
471 0543 2
472 0544 2
473 0545 1

THEN
RETURN (.STATUS);
END; ! Needs to be output
PP = .PP [PP_A_NEXT_DC8]; ! Step to next packet in chain
END; ! Loop through all pasting packets for this DCB
RETURN (SSS_NORMAL);
END; ! End of routine SMGSSREPORT_CHANGE_INSERT

		01FC 00000						.EXTRN	SMG\$_WRONUMARG		
50	58	00000000G	00	9E	00002	MOVAB	.ENTRY SMG\$REPORT_CHANGE_INSERT, Save R2,R3,R4,- : 0310				
	5E		10	C2	00009	SUBL2	R5,R6,R7,R8				
	6C		05	83	0000C	SUBB3	SMG\$MIN_UPD, R8				
	01		50	91	00010	CMPB	#16, SP				
			08	1B	00013	BLEQU	#5 (AP), DIFF				
			50	00000000G	8F	DO 00015	MOVL	DIFF, #1			
						04 0001C	RET	1\$			
						1\$:	MOVL	#SMG\$_WRONUMARG, R0			
						04 00021	CMPL	0310			
						06 12 00026	BNEQ	#DISPLAY_ID, R0			
04	11	44	A0	91	00028	CMPB	56(R0), #DISPLAY_ID				
			08	13	0002C	BEQL	2\$				
			50	00000000G	8F	DO 0002E	MOVL	68(R0), #17			
						04 00035	RET	3\$			
						04 00036	MOVL	#SMG\$_INVDIS_ID, R0			
						3\$:	RET	0417			
						04 0003A	MOVL	#DISPLAY_ID, DCB			
						11 19 0003E	MOVL	#PASTEBOARD_ID, R0			
						50 D1 00040	BLSS	4\$			
						08 14 00047	CMPL	R0, PBD_L_COUNT			
08	00	00000000G	00	50	E0 00049	BGTR	4\$				
				50	DO 00051	BBS	R0, PBD_V_PB_AVAIL, 5\$				
					4\$:	MOVL	#SMG\$_INVPAS_ID, R0				
						04 00058	RET	0418			
						52 00000000G0040	MOVL	PBD_A_PBCB[R0], PBCB			
						04 AE 9F 00061	PUSHAB	PP			
						52 DD 00064	PUSHL	PBCB			
						54 DD 00066	PUSHL	DCB			
						03 FB 00068	CALLS	#3, SMG\$LOCATE_PP			
						50 E9 0006F	BLBC	STATUS, 10\$			
66	02	A4	10	AC	DO 00072	MOVL	CHANGED_ROW, R6				
				66	D5 00076	TSTL	(R6)				
				08	15 00078	BLEQ	6\$				
				00	ED 0007A	CMPZV	#0, #16, 2(DCB), (R6)				
				08	18 00080	BGEQ	7\$				
				50	00000000G	DO 00082	MOVL	#SMG\$_INVROW, R0			
						6\$:	RET	0419			
						04 00089	MOVL	#SMG\$_INVROW, R0			
				57	14	AC DO 0008A	CHANGED_COL, R7	0423			
						7\$:	TSTL	(R7)			
67	06	A4	10	00	ED 00092	BLEQ	8\$				
				08	18 00098	CMPZV	#0, #16, 6(DCB), (R7)				
						BGEQ	9\$				
				50	00000000G	8F DO 0009A	MOVL	#SMG\$_INVCOL, R0			
					8\$:						

SMGSSPRVINP
1-001SMGSSPRVINP - Private Input support routines
SMGSSREPORT_CHANGE_INSERT - Report change to sc

M 11

16-Sep-1984 01:10:09
14-Sep-1984 13:09:59VAX-11 Bliss-32 V4.0-742
[SMGRTL.SRC]SMGPRVINP.B32;1Page 13
(6)S
1

08	AE	010E0001	8F	DD	000A2	9\$:	04	000A1	RET	0428
0C	AE	0C	AC	DD	000AA		MOVL	#17694721, DESC	0431	
		08	AE	9F	000AF		MOVL	CHANGED_CHAR, DESC+4	0441	
		7E	04	56	7D	000B2	PUSHAB	DESC	0441	
				04	AC	000B5	MOVQ	R6, -(SP)	0448	
00000000G	00	00	04	FB	000B8		PUSHL	DISPLAY_ID		
42			50	E9	000BF		CALLS	#4, SMG\$INSERT_CHARS		
06			6C	91	000C2		BLBC	STATUS, 12\$		
			24	1F	000C5		CMPB	(AP), #6		
			18	AC	D5	000C7	BLSSU	11\$		
				1F	13	000CA	TSTL	24(AP)		
04	AE	30	A4	9A	000CC		BEQL	11\$		
		0D0A	8F	3C	000D0		MOVZBL	48(DCB), -(SP)	0464	
		04	AE	9F	000D6		MOVZWL	#3338, 4(SP)	0463	
			01	DD	000D9		PUSHAB	4(SP)		
		7E	2E	A4	9A	000DB	PUSHL	#1	0459	
00000000G	00	05	54	DD	000DF		MOVZBL	46(DCB), -(SP)	0461	
5F		50	FB	000E1			PUSHL	DCB	0460	
53		04	E9	000E8	10\$:		CALLS	#5, SMG\$PUT_TEXT_TO_BUFFER	0459	
			50	DD	000EB	11\$:	BLBC	STATUS, 15\$	0473	
00000000G	00	01	53	DD	000EF		MOVL	PP, R3		
7D		50	FB	000F1			PUSHL	R3		
		53	E9	000F8			CALLS	#1, SMG\$MOVE_TEXT_TO_WINDOW_BUF	0480	
00000000G	00	01	53	DD	000FB		BLBC	STATUS, 19\$		
		71	50	FB	000FD		PUSHL	R3		
			50	E9	00104	12\$:	CALLS	#1, SMG\$MOVE_TEXT_TO_SCREEN_BUF		
			1C	A3	B5	00107	BLBC	STATUS, 19\$		
				15	13	0010A	TSTW	28(R3)		
00A8	C2	66	B0	0010C			BEQL	13\$	0487	
00AA	C2	66	B0	00111			MOVW	(R6), 168(PBCB)	0491	
00AC	C2	67	B0	00116			MOVW	(R6), 170(PBCB)	0492	
00AE	C2	5A	A2	B0	0011B		MOVW	(R7), 172(PBCB)	0495	
			52	DD	00121	13\$:	MOVW	90(PBCB), 174(PBCB)	0496	
04	AE	68	01	FB	00123		PUSHL	PBCB	0499	
		20	A4	DD	00126		CALLS	#1, SMG\$MIN_UPD		
		53	04	AE	DD	00128	MOVL	32(DCB), PP	0509	
		51	20	A4	9E	0012F	MOVL	PP, R3	0510	
		51		53	D1	00133	MOVAB	32(DCB), R1		
				3D	13	00136	CMPL	R3, R1		
		55	14	A3	DD	00138	BEQL	18\$		
		52		55	D1	0013C	MOVL	20(R3), NEW_PBCB	0516	
				2E	13	0013F	CMPL	NEW_PBCB, PBCB	0517	
00000000G	00	55	DD	00141			BEQL	17\$		
		2B	01	FB	00143		PUSHL	NEW_PBCB	0520	
			50	E9	0014A	15\$:	CALLS	#1, SMG\$FILL_WINDOW_BUFFER		
			1C	A3	B5	0014D	BLBC	STATUS, 19\$		
				15	13	00150	TSTW	28(R3)		
00A8	C2	66	B0	00152			BEQL	16\$	0524	
00AA	C2	66	B0	00157			MOVW	(R6), 168(PBCB)	0528	
00AC	C2	67	B0	0015C			MOVW	(R6), 170(PBCB)	0529	
00AE	C2	5A	A2	B0	00161		MOVW	(R7), 172(PBCB)	0532	
			55	DD	00167	16\$:	MOVW	90(PBCB), 174(PBCB)	0533	
04	AE	68	01	FB	00169		PUSHL	NEW_PBCB	0536	
		09	50	E9	0016C		CALLS	#1, SMG\$MIN_UPD		
		04	AE	63	DD	0016F	BLBC	STATUS, 19\$	0541	
				86	11	00173	MOVL	PP, R3	0510	
							BRB	14\$		

SMG\$SPRVINP
1-001

SMG\$SPRVINP - Private input support routines
SMG\$REPORT_CHANGE_INSERT - Report change to sc

N 11

16-Sep-1984 01:10:09

VAX-11 Bliss-32 V4.0-742
[SMGRTL.SRC]SMGPRVINP.B32;1

Page 14
(6)

50

01 00 00175 18\$: MOVL #1, R0
04 00 00178 19\$: RET

: 0544
: 0545

; Routine Size: 377 bytes. Routine Base: _SMG\$CODE + 005D

: 474 0546 1 !<BLF/PAGE>

```
476 0547 1 %SBTTL 'SMG$REPORT_CHANGE_REPLACE - Report change to screen -- replace'  
477 0548 1 GLOBAL ROUTINE SMG$REPORT_CHANGE_REPLACE (   
478 0549 1 DISPLAY_ID,  
479 0550 1 PASTEBOARD_ID,  
480 0551 1 NUM_CHARS : REF VECTOR [,WORD],  
481 0552 1 CHANGED_CHARS : REF VECTOR [,BYTE],  
482 0553 1 CHANGED_ROW,  
483 0554 1 CHANGED_COL,  
484 0555 1 TERMINATING_CHAR  
485 0556 1 ) =  
486 0557 1 ++  
487 0558 1 FUNCTIONAL DESCRIPTION:  
488 0559 1  
489 0560 1 This routine is called to report a change made to the physical  
490 0561 1 screen via an input action involving replacement. The  
491 0562 1 characters identified by CHANGED_CHARS and NUM_CHARS have been  
492 0563 1 positioned on the screen in a position corresponding to the  
493 0564 1 virtual display coordinates CHANGED_ROW and CHANGED_COL.  
494 0565 1  
495 0566 1 TERMINATING_CHAR specifies the details of the change.  
496 0567 1  
497 0568 1 If TERMINATING_CHAR has not been specified, then the changed  
498 0569 1 characters have been echoed in the position indicated. No  
499 0570 1 further output is required. Just need to update internal data  
500 0571 1 bases to reflect that change. Current cursor position is  
501 0572 1 changed to reflect that beyond the changed text. If this new  
502 0573 1 position is now beyond the bounds of the virtual display's  
503 0574 1 dimensions then ????.  
504 0575 1  
505 0576 1 If TERMINATING_CHAR was specified, the changed characters have  
506 0577 1 been echoed in the positions indicated. Furthermore, these  
507 0578 1 character were followed by a TERMINATING_CHAR. Just need to  
508 0579 1 update internal data bases to reflect that change. Current  
509 0580 1 cursor position is changed to reflect the effects of the  
510 0581 1 TERMINATING CHAR. If this new position is now beyond the  
511 0582 1 bounds of the virtual display's dimensions then ????.  
512 0583 1  
513 0584 1 CALLING SEQUENCE:  
514 0585 1  
515 0586 1 ret_status.wlc.v = SMG$REPORT_CHANGE_REPLACE (   
516 0587 1 DISPLAY_ID.rl.r  
517 0588 1 PASTEBOARD_ID.rl.r,  
518 0589 1 NUM_CHARS.rwu.r,  
519 0590 1 CHANGED_CHARS.rab.r  
520 0591 1 [,CHANGED_ROW.rl.r]  
521 0592 1 [,CHANGED_COL.rl.r]  
522 0593 1 [,TERMINATING_CHAR.rb.r])  
523 0594 1  
524 0595 1 FORMAL PARAMETERS:  
525 0596 1 DISPLAY_ID.rl.r Display id of virtual display.  
526 0597 1 PASTEBOARD_ID.rl.r Pasteboard id.  
527 0598 1 NUM_CHARS.rwu.r The number of characters that changed.  
528 0599 1 CHANGED_CHARS.rab.r Address of the characters that modified  
529 0600 1 the screen.  
530 0601 1  
531 0602 1  
532 0603 1
```

```

533 0604 1 | CHANGED_ROW.rl.r Row number within the virtual display in
534 0605 1 | which CHANGED_CHARS were written.
535 0606 1 |
536 0607 1 |
537 0608 1 | CHANGED_COL.rl.r Column number within the virtual display
538 0609 1 | where CHANGED_CHARS were written.
539 0610 1 |
540 0611 1 | [,TERMINATING_CHAR.rb.r] [Optional].
541 0612 1 | If supplied, the terminating character
542 0613 1 | that followed CHANGED_CHAR (See
543 0614 1 | functional description for meaning).
544 0615 1 |
545 0616 1 | IMPLICIT INPUTS:
546 0617 1 | NONE
547 0618 1 |
548 0619 1 |
549 0620 1 | IMPLICIT OUTPUTS:
550 0621 1 | NONE
551 0622 1 |
552 0623 1 |
553 0624 1 | COMPLETION STATUS:
554 0625 1 |
555 0626 1 | SSS_NORMAL Normal successful completion
556 0627 1 | SMG$_INVDIS_ID Invalid Display Id
557 0628 1 | SMG$_INVPAS_ID Invalid Pasteboard Id
558 0629 1 | SMG$_INVROW Invalid row specified
559 0630 1 | SMG$_INVCOL Invalid column specified
560 0631 1 |
561 0632 1 | SIDE EFFECTS:
562 0633 1 |
563 0634 1 | NONE
564 0635 1 | !-- 
565 0636 1 |
566 0637 2 | BEGIN
567 0638 2 | BUILTIN
568 0639 2 | NULLPARAMETER;
569 0640 2 |
570 0641 2 | LOCAL
571 0642 2 | STATUS, ! Status of subroutine calls
572 0643 2 | C_ROW, ! Working row
573 0644 2 | C_COL, ! Working col
574 0645 2 | DCB : REF $DCB DECL, ! Addr of display control block
575 0646 2 | PBCB : REF BLOCK [,BYTE], ! Addr of pasteboard control
576 0647 2 | block.
577 0648 2 | WCB : REF BLOCK [,BYTE], ! Address of window block
578 0649 2 | PP : REF BLOCK [,BYTE]; ! Addr of pasting packet.
579 0650 2 |
580 0651 2 | $SMG$VALIDATE_ARGCOUNT (4, 7); ! Test for right no. of args
581 0652 2 |
582 0653 2 | !+ Get addresses of control blocks needed.
583 0654 2 | - Get addresses of control blocks needed.
584 0655 2 |
585 0656 2 | $SMG$GET_DC B (.DISPLAY_ID, DCB); ! Get DCB addr.
586 0657 2 | $SMG$GET_PBCB (.PASTEBOARD_ID, PBCB); ! Get PBCB addr.
587 0658 2 | IF NOT (STATUS = SMG$LOCATE_PP (.DCB, .PBCB, PP)) ! Get PP addr.
588 0659 2 | THEN
589 0660 2 | RETURN (.STATUS);

```

```
590 0661 2
591 0662 2
592 0663 2 IF NOT NULLPARAMETER (5)
593 0664 3 THEN
594 0665 3 BEGIN
595 0666 3 C_ROW = ..CHANGED_ROW;
596 0667 3 DCB [DCB_W_CURSOR_ROW] = .C_ROW;
597 0668 3 END
598 0669 2 ELSE
599 0670 2 C_ROW = .DCB [DCB_W_CURSOR_ROW];
600 0671 2
601 0672 2 IF NOT NULLPARAMETER (6)
602 0673 3 THEN
603 0674 3 BEGIN
604 0675 3 C_COL = ..CHANGED_COL;
605 0676 3 DCB [DCB_W_CURSOR_COL] = .C_COL;
606 0677 3 END
607 0678 2 ELSE
608 0679 2 C_COL = .DCB [DCB_W_CURSOR_COL];
609 0680 2 $SMG$VALIDATE_ROW_COL (.C_ROW, .C_COL); ! Valid posit?
610 0681 2
611 0682 2 !+ Reflect this change in the virtual display text and attribute buffers.
612 0683 2 !-
613 0684 2
614 0685 2
615 0686 2 !+
616 0687 2 ! Invalidate physical cursor position.
617 0688 2 ! This will force output to begin with a direct cursor
618 0689 2 ! movement to the proper place.
619 0690 2 !-
620 0691 2
621 0692 2 WCB = .PBCB [PBCB_A_WCB];
622 0693 2 WCB [WCB_W_OLD_CUR_ROW] = 0;
623 0694 2 WCB [WCB_W_OLD_CUR_COL] = 0;
624 0695 2
625 0696 3 IF NOT (STATUS = SMG$PUT_TEXT_TO_BUFFER (
626 0697 3 .DCB,
627 0698 3 .DCB [DCB_B_DEF_VIDEO_ATTR],
628 0699 3 .NUM_CHARS[0],
629 0700 3 .CHANGED_CHARS,
630 0701 3 .DCB [DCB_B_DEF_CHAR_SET]))
631 0702 2
632 0703 2
633 0704 2
634 0705 2
635 0706 2 IF NOT NULLPARAMETER (7)
636 0707 3 THEN BEGIN ! Terminator supplied
637 0708 3 !+
638 0709 3 ! Inspect supplied terminator to determine effect on cursor
639 0710 3 ! position in virtual display.
640 0711 3 !-
641 0712 4 IF NOT (STATUS = SMG$PUT_TEXT_TO_BUFFER (
642 0713 4 .DCB,
643 0714 4 .DCB [DCB_B_DEF_VIDEO_ATTR],
644 0715 4 ! %REF (CR^8 + LF), ! <CR><LF>
645 0716 4 .DCB [DCB_B_DEF_CHAR_SET]))
646 0717 4
```

```
647 0718 3      THEN
648 0719 3      RETURN (.STATUS);
649 0720 2      END; ! Terminator supplied
650
651 0722 2      + Reflect this change in the appropriate positions of the window
652 0723 2      text and attribute buffers, including new screen cursor position.
653 0724 2      - IF NOT (STATUS = SMGSSMOVE_TEXT_TO_WINDOW_BUF ( .PP))
654 0725 2      THEN
655 0726 3      RETURN (.STATUS);
656 0727 2
657 0728 2      + Update screen buffers as well.
658 0729 2      - IF NOT (STATUS = SMGSSMOVE_TEXT_TO_SCREEN_BUF ( .PP))
659 0730 2      THEN
660 0731 2      RETURN (.STATUS);
661 0732 2
662 0733 3      + If this virtual display is pasted to pasteboards other than the one
663 0734 2      identified in the call list, these additional pasteboard's window
664 0735 2      buffers must be updated as well. For these additional pasteboards,
665 0736 2      the changed bytes must be output -- since they did not receive the
666 0737 2      originally echoed characters.
667 0738 2      - PP = .DCB [DCB_A_PP_NEXT]; ! 1st in chain
668 0739 2      WHILE .PP NEQ DCB [DCB_A_PP_NEXT] ! While any packets remain...
669 0740 2      DO
670 0741 2      BEGIN ! Loop through all pasting packets for this DCB
671 0742 2      LOCAL
672 0743 2      NEW_PBCB : REF $PBCB_DL_L; ! PBCB being considered
673 0744 2
674 0745 2
675 0746 2
676 0747 3      NEW_PBCB = .PP [PP_A_PBCB_ADDR]; ! PBCB for this packet
677 0748 3      IF .NEW_PBCB NEQ .PBCB ! If this isn't the one we started with
678 0749 3      THEN
679 0750 3      BEGIN ! Needs to be output
680 0751 3      IF NOT (STATUS = SMGSSFILL_WINDOW_BUFFER (.NEW_PBCB))
681 0752 3      THEN
682 0753 3      RETURN (.STATUS);
683 0754 4
684 0755 5      IF .PP [PP_W_ROWS_TO_MOVE] NEQ 0
685 0756 4
686 0757 4      THEN
687 0758 4
688 0759 4
689 0760 4      THEN
690 0761 5      BEGIN
691 0762 5      ! Assume damage confined to single row.
692 0763 5      NEW_PBCB [PBCB_W_FIRST_CHANGED_ROW] = .C_ROW;
693 0764 5      NEW_PBCB [PBCB_W_LAST_CHANGED_ROW] = .C_ROW;
694 0765 5
695 0766 5      ! Assume damage in row from given pos. to end of line
696 0767 5      NEW_PBCB [PBCB_W_FIRST_CHANGED_COL] = .C_COL;
697 0768 5      NEW_PBCB [PBCB_W_LAST_CHANGED_COL] =
698 0769 5      .NEW_PBCB [PBCB_W_WIDTH];
699 0770 4      END;
700 0771 4
701 0772 5      IF NOT (STATUS = SMGSSMIN_UPD (.NEW_PBCB))
702 0773 4      THEN
703 0774 4      RETURN (.STATUS);
```

```

: 704 0775 4
: 705 0776 3
: 706 0777 3
: 707 0778 3      END;      ! Needs to be output
: 708 0779 2      PP = .PP [PP_A_NEXT_DC8]; ! Step to next packet in chain
: 709 0780 2      END;      ! Loop through all pasting packets for this DCB
: 710 0781 2      RETURN (SSS_NORMAL);
: 711 0782 1      END;      ! End of routine SMGSSREPORT_CHANGE_REPLACE

```

		01FC 00000	.ENTRY	SMGSSREPORT_CHANGE_REPLACE, Save R2,R3,R4,-	0548	
50	58 00000000G	00 9E 00002	MOVAB	R5,R6,R7,R8		
	5E	08 C2 00009	SUBL2	SMGSSPUT_TEXT_TO_BUFFER, R8		
	6C	04 83 0000C	SUBB3	#8, SP		
	03	50 91 00010	CMPB	#4, (AP), DIFF	0651	
		08 18 00013	BLEQU	DIFF, #3		
		50 00000000G	00 00015	MOVL	1\$	
		8F	04 0001C	RET	#SMGS_WRONUMARG, R0	
	04	50 04 BC D0 0001D	1\$:	MOVL	0DISPLAY_ID, R0	
		BC 38 A0 D1 00021		CMPL	56(R0), 0DISPLAY_ID	0656
		06 12 00026		BNEQ	2\$	
11	44 A0 91 00028		CMPB	68(R0), #17		
	08 13 0002C		BEQL	3\$		
	50 00000000G	00 0002E	MOVL	#SMGS_INVDIS_ID, R0		
	8F	04 00035	RET			
53	04 BC D0 00036	3\$:	MOVL	0DISPLAY_ID, DCB		
	50 08 BC D0 0003A		MOVL	0PASTEBOARD_ID, R0	0657	
	11 19 0003E		BLSS	4\$		
00000000G	00 50 D1 00040		CMPL	R0, PBD_L_COUNT		
	08 14 00047		BGTR	4\$		
08 00000000G	00 50 E0 00049		BBS	R0, PBD_V_PB_AVAIL 5\$		
	50 00000000G	8F D0 00051	4\$:	MOVL	#SMGS_INVPAS_ID, R0	
	04 00058		RET			
	55 00000000G0040	00 00059	5\$:	MOVL	PBD_A_PBCB[R0], PBCB	
	04 AE 9F 00061		PUSHAB	PP	0658	
00000000G	28 BB 00064		PUSHR	#^M<R3,R5>		
	00 03 FB 00066		CALLS	#3, SMGSSLOCATE_PP		
	73 50 E9 0006D		BLBC	STATUS, 14\$		
	05 6C 91 00070		CMPB	(AP), #5	0662	
		0F 1F 00073		BLSSU	6\$	
		14 AC D5 00075		TSTL	20(AP)	
		0A 13 00078		BEQL	6\$	
	28 A3 14	BC D0 0007A		MOVL	0CHANGED_ROW, C_ROW	0665
		56 B0 0007E		MOVW	C_ROW, 40(DCB)	0666
	06 56	04 11 00082		BRB	7\$	0662
	28 A3 3C 00084	6\$:	MOVZWL	40(DCB), C_ROW	0669	
06	6C 91 00088	7\$:	CMPB	(AP), #6	0671	
	0F 1F 00088		BLSSU	8\$		
	18 AC D5 0008D		TSTL	24(AP)		
	0A 13 00090		BEQL	8\$		
2A A3 18	BC D0 00092		MOVL	0CHANGED_COL, C_COL	0674	
	57 B0 00096		MOVW	C_COL, 42(DCB)	0675	
	04 11 0009A		BRB	9\$	0671	

SMGSSPRVNP
1-001

SMGSSPRVINP - Private Input support routines
SMGSSREPORT_CHANGE_REPLACE - Report change to

G 12

16-Sep-1984 01:10:09
14-Sep-1984 13:09:59

VAX-11 Bliss-32 V4.0-742
[SMGRTL.SRC]SMGPRV1NP.B32:1

Page 20
(7)

S1

56	02	A3	57	2A	A3	3C	0009C	8\$:	MOVZWL	42(DCB), C_COL	0678
					56	D5	000A0	9\$:	TSTL	C_ROW	0680
			10		08	15	000A2		BLEQ	10\$	
					08	18	000AA		CMPZV	#0, #16, 2(DCB), C_ROW	
			50 00000000G	8F	DD	000AC	10\$:	BGEQ	11\$		
					04	000B3		MOVL	#SMGS_INVROW, R0		
					57	D5	000B4	11\$:	RET		
					08	15	000B6		TSTL	C_COL	
			10		00	ED	000B8		BLEQ	12\$	
					08	18	000BE		CMPZV	#0, #16, 6(DCB), C_COL	
			50 00000000G	8F	DD	000C0	12\$:	BGEQ	13\$		
					04	000C7		MOVL	#SMGS_INVCOL, R0		
					51	08	A5	13\$:	RET		
					24	A1	D4		MOVL	8(PBCB), WCB	0696
			7E		30	A3	9A		CLRL	36(WCB)	0697
					10	AC	DD		MOVZBL	48(DCB), -(SP)	0701
			7E		0C	BC	3C		PUSHL	CHANGED_CHARS	0700
			7E		2E	A3	9A		MOVZWL	2NUM_CHARS, -(SP)	0699
					53	DD	000D6		MOVZBL	46(DCB), -(SP)	0698
					68	05	FB		PUSHL	DCB	0697
			63		50	E9	000E3	14\$:	CALLS	#5, SMGSSPUT_TEXT_TO_BUFFER	
			07		6C	91	000E6		BLBC	STATUS, 17\$	0696
					2C	1F	000E9		(AP), #7		0705
					1C	AC	D5		BLSSU	15\$	
			04	7E	30	A3	9A		TSTL	28(AP)	
				AE	0D0A	8F	3C		BEQL	15\$	
					04	AE	9F		MOVZBL	48(DCB), -(SP)	0717
					01	DD	000F0		MOVZWL	#3338, 4(SP)	0716
			7E		2E	A3	9A		PUSHAB	4(SP)	
					53	DD	000F4		PUSHL	#1	0712
					68	05	FB		MOVZBL	46(DCB), -(SP)	0714
			70		50	E9	000FF		PUSHL	DCB	0713
					70	04	AE	16\$:	CALLS	#5, SMGSSPUT_TEXT_TO_BUFFER	
			00000000G	00	01	FB	00105		BLBC	STATUS, 21\$	0712
				63	50	E9	00108		PUSHL	PP	0726
			00000000G	00	04	AE	DD	15\$:	CALLS	#1, SMGSSMOVE_TEXT_TO_WINDOW_BUF	
				56	01	FB	00115		BLBC	STATUS, 21\$	
			04	AE	DD	00118		PUSHL	PP	0733	
				01	50	E9	00122		CALLS	#1, SMGSSMOVE_TEXT_TO_SCREEN_BUF	
			04	AE	20	A3	DD		BLBC	STATUS, 21\$	
				54	04	AE	00125		MOVL	32(DCB), PP	0744
			04	AE	20	A3	9E	16\$:	MOVL	PP, R4	0745
				51	20	9E	0012E		MOVAB	32(DCB), R1	
				51	54	D1	00132		CMPL	R4, R1	
				52	41	13	00135		BEQL	20\$	
			52	14	A4	DD	00137		MOVL	20(R4), NEW_PBCB	0751
				55	52	D1	00138		CMPL	NEW_PBCB, PBCB	0752
				32	32	13	0013E		BEQL	19\$	
			00000000G	00	52	DD	00140		PUSHL	NEW_PBCB	0755
				2F	01	FB	00142		CALLS	#1, SMGSSFILL_WINDOW_BUFFER	
					50	E9	00149	17\$:	BLBC	STATUS, 21\$	
					1C	A4	B5		TSTW	28(R4)	0759
					15	13	0014F		BEQL	18\$	
			00A8	C2	56	B0	00151		MOVW	C_ROW, 168(NEW_PBCB)	0763
			00AA	C2	56	B0	00156		MOVW	C_ROW, 170(NEW_PBCB)	0764
			00AC	C2	57	B0	00158		MOVW	C_COL, 172(NEW_PBCB)	0767
			00AE	C2	5A	A2	B0		MOVW	90(NEW_PBCB), T74(NEW_PBCB)	0769

SMGSSPRVINP
1-001

SMGSSPRVINP - Private Input support routines
SMGSSREPORT_CHANGE_REPLACE - Report change to s

H 12

16-Sep-1984 01:10:09
14-Sep-1984 13:09:59

VAX-11 Bliss-32 V4.0-742
[SMGRTL.SRC]SMGPRVINP.B32;1

Page 21
(?)

00000000G	00	52 DD 00166 18\$:	PUSHL	NEW_PBCB	: 0772
	09	01 F8 00168	CALLS	#1-SMGSSMIN_UPD	
	04	50 E9 0016F	BLBC	STATUS, 21\$	
	AE	64 D0 00172 19\$:	MOVL	(R4), PP	: 0778
	50	B2 11 00176	BRB	16\$: 0745
		01 D0 00178 20\$:	MOVL	#1, R0	: 0781
		04 0017B 21\$:	RET		: 0782

; Routine Size: 380 bytes, Routine Base: _SMGSCODE + 0106

: 712 0783 1 !<BLF/PAGE>

SMGSSPRVINP
1-001

SMGSSPRVINP - Private Input support routines
SMGSSREPORT_CHANGE_REPLACE - Report change to s

112

16-Sep-1984 01:10:09

14-Sep-1984 13:09:59

VAX-11 Bliss-32 V4.0-742
[SMGRTL.SRC]SMGPRVINP.B32;1

Page 22
(8)

: 714 0784 1 END
: 715 0785 1
: 716 0786 0 ELUDOM

: ! End of module SMGSSPRVINP

PSECT SUMMARY

Name	Bytes	Attributes
_SMG\$CODE	850	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	-----	Symbols	-----	Pages	Processing
	Total	Loaded	Percent	Mapped	Time
-\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	9	0	581	00:01.0
-\$255\$DUA28:[SMGRTL.OBJ]RTLLIB.L32;1	36	0	0	8	00:00.1
-\$255\$DUA28:[SMGRTL.OBJ]SMGLIB.L32;1	469	38	8	38	00:00.4

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:SMGPRVINP/OBJ=OBJ\$:SMGPRVINP MSRC\$:SMGPRVINP/UPDATE=(ENH\$:SMGPRVINP
)

: Size: 850 code + 0 data bytes
: Run Time: 00:18.4
: Elapsed Time: 01:10.5
: Lines/CPU Min: 2557
: Lexemes/CPU-Min: 16151
: Memory Used: 162 pages
: Compilation Complete

0360 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

